Flask – aplikacja ToDo (cz. 2)

Aplikacja ToDo ma służyć do tworzenia, przeglądania, edycji i usuwania listy zadań do wykonania.

Masz już model danych (klasę Zadanie), widok domyślny i widok wyświetlający treść zadań. Rozwijasz aplikację o możliwości dodawania, usuwania zadań oraz oznaczania jako zrobione.

I. W pliku `models.py`:
   1. Do definicji klasy *Zadanie* dodaj klasę *Meta* z opcją **`order_by = ['data_dodania']`**.

II. Wykorzystując szablon `szkielet.html`:
   1. Utwórz szablon `index.html`, zawierający tekst zwracany przez widok **`index()`**.
   2. Zmień szablon `zadania.html` tak, aby wykorzystywał szablon bazowy.

III. W pliku `app.py`:
   1. Zmień widok domyślny tak, aby zwracał szablon `index.html`.

IV. W pliku `zadania.html`:
   1. Uzupełnij kod formularza umożliwiającego dodawanie zadań. Powinien zawierać tylko jedno pole przeznaczone na treść zadania i kierować pod adres obsługiwany przez widok **`zadania()`**.
   2. Po każdym zadaniu i dacie dodaj linki/przyciski o treści "Zrobione" i "Usuń" kierujące pod adresy obsługiwane odpowiednio przez widoki **`zrobione()`** oraz **`usun()`**.

V. W pliku `app.py`:
   1. Uzupełnij kod widoku **`zadania()`** tak, aby obsłużyć żądania typu POST, za pomocą których użytkownik może dodawać zadania. Kod powinien sprawdzać poprawność przesłanych danych, zapisywać nowe zadania w bazie oraz zwracać komunikaty zwrotne użytkownikowi i przekierowywać na adres "/zadania".
   2. Utwórz widok **`zrobione()`** powiązany z odpowiednim adresem dostosowanym do metody przesyłania identyfikatora zadania. Funkcja powinna oznaczać zadania o przesłanym identyfikatorze jako zrobione, zwracać komunikat użytkownikowi i przekierowywać pod adres "/zadania".
   3. Utwórz widok **`usun()`** powiązany z adresem z odpowiednim adresem dostosowanym do metody przesyłania identyfikatora zadania. Funkcja powinna usuwać zadania o przesłanym identyfikatorze, zwracać komunikat użytkownikowi i przekierowywać pod adres "/zadania".

Pomoc

Dodając klasę *Meta* w modelu *Zadanie* wzoruj się na definicji klasy *BazaModel*.

W szablonach zależnych od bazowego możesz używać tagów:

`{% extends nazwa_szablonu %}`

`{% block nazwa_bloku %} treść {% endblock %}`

Aby sprawdzić typ żądania, należy użyć instrukcji typu: **`if request.method == typ_żądania`**

Jeżeli użyjesz żądań GET do przesłania identyfikatora zadania, wykorzystaj: **`/zrobione/<int:zid>, def zrobione(zid), url_for('zrobione', zid=identyfikator)`**.

Jeżeli użyjesz żądań POST do przesłania identyfikatora zadania, wykorzystaj: **`/zrobione, request.form['zid'], url_for('zrobione'), <input type="hidden" name="zid" value="identyfikator">`**.